Web services – Assessment Task 3

# JAX-RS Web service

# **Description**

Create JAX-RS Web service (RESTful Web service) and explore its functionality with web services testing tools, like SoapUI or Postman applications.

# Steps

1. Create new Java project
2. Prepare the infrastructure for the task: create Java class with different field types (String, int, float, boolean, char, etc.). Add field of type List with the list of dependent objects. For example, the class Student could have a list of Subjects associated with this student.
3. Create JAX-RS Web service, use corresponding annotations.
4. Publish Web service endpoint. Run the project.
5. Open Web services testing tool SoapUI, specify the published endpoint, obtain the WADL from Web service and send the requests to the running Web service. Observe the responses.
6. Using tools like APIMATIC (see [3]), transform the WADL to OpenAPI (see [4]) description in json format.
7. Using Swagger Editor (see [5]), explore your API, send the requests and observe the responses.
8. Using Postman tool (see [6]) send the requests to created and running RESTful application and observe the responses.

# Requirements

1. The entire code should be properly formatted.
2. The package/class/field/method names should conform to the naming conventions.
3. The Unit tests for all classes should present.
4. The entire code should be properly documented and JavaDoc generated.
5. The entire code should conform to S.O.L.I.D principles.
6. The deadline is May 15.

# References

1. [Building RESTful Web Services with JAX-RS](https://docs.oracle.com/javaee/6/tutorial/doc/giepu.html)
2. <https://www.soapui.org/>
3. <https://www.apimatic.io/transformer>
4. [OpenAPI Specification](https://swagger.io/specification/)
5. <https://editor.swagger.io/>
6. <https://www.getpostman.com/>
7. [Java Naming conventions](https://www.javatpoint.com/java-naming-conventions)
8. [Naming a Package](https://docs.oracle.com/javase/tutorial/java/package/namingpkgs.html)
9. [How to Write Doc Comments for the Javadoc Tool](https://www.oracle.com/technetwork/java/javase/tech/index-137868.html)
10. [S.O.L.I.D: The First 5 Principles of Object Oriented Design](https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design)